



Objektorientierte Programmierung mit C++ (WS 2014/2015)

Abgabe bis zum 21. Oktober 2013, 14:00 Uhr

Lernziele:

- Erste Vertrautheit mit der Unix-Shell und die
- Fähigkeit, ein einfaches C++-Programm mit einem Texteditor unter Unix zu erstellen, zu übersetzen und auszuführen.

Hinweise:

Melden Sie sich für diese Veranstaltung in SLC an. Ohne diese Anmeldung können Sie keine Lösungen elektronisch einreichen. Sobald Sie sich angemeldet haben, wird es bis zu einer Stunde brauchen, bis Sie anschließend Lösungen einreichen können.

Aufgabe 1: Erste Schritte mit der Shell und mit einem Texteditor

Arbeiten Sie auf einem der Arbeitsplätze in unseren Pools oder melden Sie sich von anderswo per SSH (*secure shell*) auf *thales.mathematik.uni-ulm.de* an. Sobald die Anmeldung erfolgreich war, sollten Sie den Prompt der Shell sehen. Per Voreinstellung ist dies bei uns der jeweilige Rechnername, gefolgt von einem Dollarzeichen, also beispielsweise „*thales\$*“.

Kommandos beginnen mit dem Namen des Kommandos und weiteren Parametern. Ein einfaches Kommando ist beispielsweise *find*, mit dem Sie nach Dateien suchen können, die bestimmte Kriterien erfüllen. Das folgende Kommando sucht beispielsweise nach der Header-Datei *iostream* unterhalb der Verzeichnishierarchie */usr/local/gcc49* (dort ist auf der Thales GCC 4.9.1 installiert):

```
thales$ find /usr/local/gcc49 -name iostream  
/usr/local/gcc49/include/c++/4.9.1/iostream  
thales$
```

Wenn Sie sich über diese oder andere Kommandos näher informieren möchten, helfen die Manualseiten. Die Manualseite zu *find* können Sie mit „man -s 1 find“ abrufen. („-s 1“ steht für Sektion 1 – das sind die normalen Kommandos auf der Shell-Ebene.) Probieren Sie das aus und ermitteln Sie mit Hilfe der Manualseite, wie Sie *find* so aufrufen können, dass Sie alle für die Welt ausführbaren regulären Dateien unterhalb von */usr/local/gcc49* ermitteln.

Hinweis: Wenn Sie die Manualseite lesen, befinden Sie sich in einem Programm zum seitenweisen Lesen von Text, das sich *less* nennt. Mit der Leertaste können Sie vorwärts blättern, mit „b“ rückwärts gehen, mit „h“ erhalten Sie einen Hilfetext und mit „q“ können Sie das Lesen beenden, so dass Sie wieder in die Shell zurückkehren.

Falls Sie noch nie einen Texteditor unter Unix benutzt haben, ist es jetzt an der Zeit, dies zu lernen. Empfehlenswert und besonders geeignet für Software-Entwicklung wäre insbesondere der *vim*. Die ersten Schritte lassen sich am besten mit dem *vitutor* erlernen. Sie können aber notfalls auch auf den *pico* zurückgreifen, der von der ersten Benutzung an vollkommen selbsterklärend ist und deswegen zu Beginn etwas einfacher erscheint, obwohl er auf lange Sicht deutlich hinter den Möglichkeiten des *vim* zurückbleibt.

Erstellen Sie mit dem Texteditor eine Datei namens *find.txt* und beschreiben Sie darin, wie *find* so aufgerufen werden kann, dass die Ausgabe in der oben beschriebenen Weise erfolgt. Diese Datei können Sie danach elektronisch einreichen:

```
thales$ submit cpp 1 find.txt
thales$
```

Wenn es geklappt hat, gibt es hier keine Fehlermeldung. Wenn es mit der Registrierung bei SLC noch nicht geklappt hat, erscheint die Meldung „submit: you are not yet registered for cpp“.

Aufgabe 2: Ein erstes C++-Programm

Bevor Sie auf der Thales mit C++ beginnen, sollten Sie sicherstellen, dass Sie den neueren GCC als Übersetzer verwenden. Dazu sollte in der Datei *.options* in Ihrem Heimatkatalog eine Zeile „gcc49“ enthalten sein. Sollte dies noch nicht der Fall sein, kann eine entsprechende Zeile mit

```
thales$ echo gcc49 >> ~/.options
thales$
```

hinzugefügt werden. Danach ist ein Ab- und Anmelden auf der Thales erforderlich, damit diese Änderung erfolgreich ist. Auf die Schnelle geht dies aber auch mit folgendem Kommando:

```
thales$ . /etc/profile
[...Tagesmeldungen...]
thales$
```

Danach sollte der GCC 4.9.1 zur Verfügung stehen:

```
thales$ g++ --version | sed 1q
g++ (GCC) 4.9.1
thales$
```

Zu den ersten Versuchen in einer neuen Programmiersprache gehört typischerweise ein Programm, das einen mit einer Ausgabe begrüßt. Versuchen Sie also ein C++-Programm mit dem Dateinamen „hello.cpp“ zu erstellen, das eine beliebige Begrüßung ausgibt. Sie finden dazu Beispiele auf dem Netz, wenn Sie nach „hello world“ suchen. Es gibt mehrere Möglichkeiten dazu. Da C++ fast vollständig aufwärtskompatibel zu C ist, können Sie es ganz einfach mit einem entsprechenden C-Programmtext versuchen. Alternativ können Sie sich nach einer Variante umschaun, bei der die *iostream*-Bibliothek verwendet wird. Sie können Ihr C++-Programm mit dem Kommando `g++` übersetzen:

```
thales$ g++ -Wall -std=gnu++11 -o hello hello.cpp
```

Die Option „-Wall“ schaltet alle Warnungen ein, mit „-std=gnu++11“ wird C++11 (das ist der aktuelle offizielle C++-Standard) zugrundegelegt mitsamt einigen GNU-Erweiterungen, mit „-o hello“ wird der Name des zu erzeugenden ausführbaren Programms bestimmt. Wenn es klappt, gibt es keine Meldung (frei nach der Philosophie *no news are good news*) und die Datei *hello* ist erzeugt. Im Erfolgsfalle können Sie das erzeugte Programm ausführen, indem Sie *hello* aufrufen:

```
thales$ hello
Hello world!
thales$
```

Ihre Lösung können dann wiederum mit *submit* einreichen:

```
thales$ submit cpp 2 hello.cpp team
```

Viel Erfolg!